12-27-99

A

Case/Docket No. TRANS18
Express Mail No.EK380384375US

THE HONORABLE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of:

Inventors: Robert Bedichek, David Keppel, and John Banning

Entitled: INTERPAGE PROLOGUE TO PROTECT VIRTUAL ADDRESS MAPPINGS

Enclosed are:

XXX Two (2) sheet(s) of Formal Drawing(s) including four (4) figures.
XXX An Assignment of the invention to Transmeta Corporation on Oct 18, 1999.
XXX A Declaration and Power of Attorney.
XXX A Verified Statement to establish Small Entity Status under 37 CFR 1.9 and 37 CFR 1.27.
XXX Return addressed stamped postcard.

The Filing Fee has been calculated as shown below:

	(Col. 1)	(Col.2)	SMALL ENTITY		OTHER THAN A SMALL ENTITY	
For:	No. Filed	No. Extra	RATE	FEE	RATE	FEE
Basic Fee:	-	-	-	$380.00	-	$760.00
Total Claims:	-14	-0-	x $9.00		x $18.00	-0-
Indep. Claims:	-2	-3-	x $39.00		x $78.00	-0-
☐ Multiple Dep. Claim(s) Presented			+ $130.00		+ $260.00	-0-
* If the difference in (Col. 1) is less than zero, enter "0" in (Col. 2)			Total:	$ 380.00	Total:	$.00

XXX A check in the amount of $420.00 is enclosed to cover the filing fee and recording the assignment..

Respectfully submitted,

Date: Dec 23, 1999

Stephen L. King, Reg. No. 19,180

30 Sweetbay Road
Rancho Palos Verdes, California 90275
(310) 377-5073

Applicant or Patentee: TRANSMETA CORPORATION Attorney's
Serial or Patent No.: _____ Docket No. TRANS18
Filed or Issued: _____

For: INTERPAGE PROLOGUE TO PROTECT VIRTUAL ADDRESS MAPPINGS

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS
37 CFR 1.9 (f) and 1.27(c) - - SMALL BUSINESS CONCERN

I hereby declare that I am an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN: TRANSMETA CORPORATION
ADDRESS OF CONCERN: 3940 FREEDOM CIRCLE, SANTA CLARA, CALIFORNIA 95054

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.3-18, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby certify that to the best of my knowledge and belief rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention entitled, INTERPAGE PROLOGUE TO PROTECT VIRTUAL ADDRESS MAPPINGS,

by inventor(s) Robert Bedichek, David Keppel, and John Banning
described in
[X] the specification being filed herewith,

and I have reviewed the document that evidences the conveyance of those rights. That document
[X] is being filed herewith.

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed below and **no rights to the invention are held by any person, other than the inventor, who could not qualify as a small business concern under 37 CFR 1.9(d) or by any concern which would not qualify as a small business concern under 347 CFR 1.9(d) or a non-profit organization under 37 CFR 1.9(e)**. NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)
NAME: _____
ADDRESS: _____
[____] Individual [_____] Small Business Concern [_____] Non-Profit Organization
NAME: _____
ADDRESS: _____
[____] Individual [_____] Small Business Concern [_____] Non-Profit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: DAVID DITZEL
TITLE OF PERSON OTHER THAN OWNER: PRESIDENT & CHIEF EXECUTIVE OFFICER
ADDRESS OF PERSON SIGNING: 3940 FREEDOM WAY, SANTA CLARA, CALIFORNIA 95054
SIGNATURE: _____ DATE: 12-21-99

Our Ref: Trans18

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

INTERPAGE PROLOGUE TO PROTECT VIRTUAL ADDRESS MAPPINGS

Inventors:

Robert Bedichek
David Keppel
John Banning

Prepared by:

Stephen L. King
30 Sweetbay Road
Rancho Palos Verdes, California 90275
(310) 377-5073

Express Mail Label No. EK380384375US

INTERPAGE PROLOGUE TO PROTECT VIRTUAL ADDRESS MAPPINGS

BACKGROUND OF THE INVENTION

Field Of The Invention

This invention relates to computer systems and, more particularly, to methods and apparatus for assuring consistency of translated instructions being executed by a microprocessor which dynamically translates instructions from a target to a host instruction set.

History Of The Prior Art

Recently, a new microprocessor was developed which combines a simple but very fast host processor (called a "morph host") and software (referred to as "code morphing software") to execute application programs designed for a processor having an instruction set different than the instruction set of the morph host processor. The morph host processor executes the code morphing software which translates the application programs dynamically into host processor instructions which are able to accomplish the purpose of the original software. As the instructions are translated, they are stored in a translation buffer where they may be executed without further translation. Although the initial translation of a program is slow, once translated, many of the steps normally required for hardware to execute a program are eliminated. The new microprocessor has proven able to execute translated "target" programs as fast as the "target" processor for which the programs were designed.

The new microprocessor is described in detail in U. S. patent 5,832,205, Memory Controller For A Microprocessor For Detecting A Failure Of

Speculation On The Physical Nature Of A Component Being Addressed, Kelly et al, November 3, 1998, assigned to the assignee of the present invention.

One reason that the new processor is able to execute programs rapidly is its ability to link together sequences of translations that occur frequently into very long sequences. Linking eliminates many of the steps which would be necessary to retrieve individually the various translations for execution. The process by which this is accomplished is explained in detail in the above-mentioned patent.

One problem that must be resolved for a computer which executes host translations of a target program is that the target program typically defines the sequences of target instructions which are to be executed by presenting a series of addresses at which those instructions are stored to the central processor as those target instructions are to be executed. The central processor reads the address of the instruction next to be executed, fetches that instruction from memory, and executes the instruction. When the target program being executed is defined by such a sequence of addresses yet the instructions being executed are host translations of those instructions which reside at other addresses, it is necessary to determine that each translated host instruction is, in fact, the result of a translation from a target instruction which is at the address (including the effect of address mapping) presented by the target program for execution.

This is an especially difficult problem where sequences of translated instruction have been linked together in the manner described above in order to attain rapid execution.

It is desirable to improve the operation of a computer system which utilizes a microprocessor that translates programs dynamically from target instructions into host instructions able to accomplish the purpose of the original software by rapidly determining that a host instruction is a translation of a target instruction presented for execution.

Summary Of The Invention

It is, therefore, an object of the present invention to improve the operation of a computer system which utilizes a microprocessor to translate programs dynamically from target instructions into host microprocessor instructions able to accomplish the purpose of the original software by rapidly determining that a host instruction is a translation of a target instruction presented for execution.

This and other objects of the present invention are realized in a computer which translates instructions from a target instruction set to a host instruction set by a process for testing the memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made, executing the translation if the addresses compare, and generating an exception if the addresses do not compare.

These and other objects and features of the invention will be better understood by reference to the detailed description which follows taken

together with the drawings in which like elements are referred to by like designations throughout the several views.

Brief Description Of The Drawings

Figures 1 is a block diagram illustrating a new microprocessor which is adapted to carry out the present invention.

Figure 2 is a diagram illustrating some of the steps of a process for carrying out the invention.

Figure 3 is a diagram illustrating some of the steps of another process for carrying out the invention.

Figure 4 is another diagram illustrating the steps of a general process for carrying out the invention.

Detailed Description

Figure 1 illustrates a microprocessor 11 which may utilize the present invention. The microprocessor pictured is described in detail in U. S. patent 5,832,205. The microprocessor includes a floating point unit 12, an integer unit 13, a translation buffer 14 which is a part of system memory, target memory 15 which is another portion of system memory, and a translation lookaside buffer 18 which is a part of a memory management unit 19.

As described above, the new microprocessor translates target instructions into host instructions which it executes. The target instructions are stored at addresses in a target portion of system memory while translated host instructions are stored at different addresses in a

host portion of system memory. When a target program is being executed, an instruction pointer pointing to the address of each sequential target instruction to be executed is provided in an EIP register. The new processor reads the instruction pointer, determines the address of the target instruction, and decides whether a host translation exists for that instruction. If no translation exists, then a new translation is generated, stored in host memory, and executed.

If a translation exists, the code morphing software finds the translation and executes it. If the translation is not linked to any other translations, then after it has been executed software determines the next target instruction from the next instruction pointer (often herein referred to as the EIP value) in the EIP register, determines whether a translation exists for this next target instruction, and continues with the process in the manner just described.

On the other hand, if the translation which is being executed is linked to other translations, then the next host instruction to be executed is not determined by the target instruction EIP value. Instead, the next host instruction is determined by a jump command to the next translation placed at the end of the executing translation by a linking process of the code morphing software.

In each case in which a next host translation is to be executed, it is important for the software to ascertain that the translation which is to be executed is a translation of the target instruction commanded by the target program which is being executed by means of the host translation and that both the target instruction EIP value and the physical address it

maps to correspond and are correct. The translation to be executed must be a translation of the target instruction to which the EIP value points and which is stored at the physical address the EIP maps to if execution is to produce a correct result.

In order to be able to ascertain that the translation which is to be executed is a translation of the target instruction pointed to by the EIP value and stored at the physical address the EIP value maps to, the present invention stores for each translation the physical address in memory of the target instruction from which the translation was made, the EIP of that target instruction, and various context relating to the translation not pertinent to the present invention.

When the instruction to be executed is being "dispatched" (responding to the next target instruction of the target program), the operation proceeds in a straightforward manner as shown in Figure 2. The code morphing software simply maps the EIP value to a physical address and finds the translation with the same EIP value and physical address. To accomplish this, the value in the EIP register is furnished to the code morphing software. The code morphing software adds the code segment base and the EIP value and furnishes the result to the translation lookaside buffer circuitry. The translation lookaside buffer circuitry utilizes this logical address to determine the physical page address pointed to by the EIP value. The physical page address and the low order bits of the sum of the code segment base and the EIP value are then combined to provide the physical address of the target instruction. With this physical address, a translation is found that matches this physical address and EIP value. If such a translation is found, control is

transferred to it. If none is found, a new translation is made of the target instructions starting at that EIP value and physical address.

However, when a first host translation is linked to one or more succeeding translations, the operation is much more complicated. Although the EIP value of the first translation in the sequence will have been used to check that its physical address and the physical address stored with the first host translation are in fact the same, the next translation (translated instruction) is reached by a linking jump instruction which completes the execution of the first translation. Since such a jump instruction does not return control to the dispatching portion of the program but rather intentionally eliminates that portion of the program in order to attain speed, no check would normally be made to determine that the host instruction is a translation of the target instruction stored at the memory address pointed to by the EIP value at the physical address mapped to by the EIP value.

Because the EIP value is changed in a fixed and predictable manner, it can be guaranteed that the EIP value of the second translation is correct when the two translations are chained together. However, the mapping of the EIP value of the second translation to a physical address may not be the same as the mapping of the EIP value of the first translation to a physical address so something must be done to assure that the mapping of the EIP value of the second translation is to the physical address of the second translation.

The present invention assures that such check is, in fact, carried out by providing a process for accomplishing the consistency test in all

situations where required. The jump instruction which accomplishes the linking is caused to point to a process which accomplishes the check and then moves to the linked translation if the test is met. If the test is not met, an exception is generated; and various solutions may be provided. It should be noted that the process of the present invention also provides a solution to situations in which an operating system has for some reason unmapped a particular memory page at which translated target instruction are stored. The prologue process will indicate a failure when its test is conducted so that the link to the next translation will not be executed. At that point, the code morphing software may roll back execution to the last consistent point from which some other method for executing the target instructions may take place.

In one embodiment of the invention, the process (shown in Figure 3) for accomplishing the check utilizes the EIP value designating the next target instruction and the code segment base to determine the virtual address pointed to by the EIP value, determines the physical address of the target instruction by a lookup, and then compares this physical address with the physical address of the target instruction stored for the translation. If the addresses are the same, the linked translation is executed. If the addresses differ, an exception is generated; and execution is rolled back to the last point at which consistent state existed and some other method of executing the target instruction is undertaken. At that point, the succeeding translation can be invalidated; and another translation or an interpretation of the target instructions can be undertaken. Alternatively, the succeeding translation may remain valid, but its link to the preceding translation

broken so that the first translation no longer proceeds directly to the second.

The need for this general test may be eliminated if the EIP value for the succeeding translation maps to an address on the same memory page as the EIP stored for the preceding translation because the consistency test will already have been conducted for the memory page and the prior translation will not have changed the mapping. If the logical addresses are on the same memory page, then the physical addresses are also on the same memory page. The information to determine this is available from the EIP values of the preceding and succeeding translations and the physical address of the first target instruction of the preceding translations. The EIP value furnished by the target program and the physical address provide sufficient information to determine whether the stored EIP values saved for the host translation are on the same memory page.

The process of the present invention may be utilized in at least two distinct ways which are both shown in figure 3 of the drawing. If it is known that the translation will be linked to a previous translation when it is translated and a check of the physical address is required, then the prologue process (described above and illustrated in Figure 3) may be included in the translation. In such as case, the preceding translation merely jumps to the next translation where the address consistency process of the prologue is executed before the translation is executed.

If, on the other hand, it is not known when the translation is made whether it will be linked to another translation or not, then the prologue

process is generated as a separate short prologue when the linking of the two translations occurs. At this time, the preceding translation is provided a jump instruction to the prologue process; and the prologue completes (if the test is met) with another jump instruction to the succeeding translation.

Another manner of utilizing the prologue process would be to incorporate what started as a separate prologue into a linked translation at the time of linking.

A single translation may include translations from target code which appears on two different pages. In this case, the translation must include code for performing the EIP mapping and physical address consistency check described above for target instructions on any page other than the page of the first target instruction.

Although the present invention has been described in terms of a preferred embodiment, it will be appreciated that various modifications and alterations might be made by those skilled in the art without departing from the spirit and scope of the invention. The invention should therefore be measured in terms of the claims which follow.

What Is Claimed Is:

Claim 1. In a computer which translates instructions from a target instruction set to a host instruction set, a method for determining validity of a translation of a target instruction linked to an earlier translation comprising the steps of:

testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made,

executing the translation if the addresses compare, and

generating an exception if the addresses do not compare.

Claim 2 A method as claimed in Claim 1 in which the step of testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made is a process separate from the translation of the target instruction.

Claim 3. A method as claimed in Claim 1 in which the step of testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made is included as a part of the translation of the target instruction.

Claim 4. A method as claimed in Claim 1 which includes an additional step of copying a memory address of a target instruction when a translation of the target instruction is made and linked to an earlier translation.

Claim 5. A method as claimed in Claim 1 which includes additional steps of copying a memory address of a target instruction when a translation of the target instruction is made, and

storing the memory address of a target instruction for comparison with a memory address of a target instruction to be executed.

Claim 6. A method as claimed in Claim 1 which includes the additional step of executing the translation without testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made if testing can be safely eliminated.

Claim 7. A method as claimed in Claim 1 which includes the additional step of executing the translation without testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made if the memory addresses are on the same memory page.

Claim 8. Computer implemented software means for determining validity of a translation of a target instruction linked to an earlier translation in a computer which translates instructions from a target instruction set to a host instruction set comprising:

means for testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made,

means for executing the translation if the addresses compare, and

means for generating an exception if the addresses do not compare.

Claim 9. Computer implemented software means as claimed in Claim 8 in which the means for testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made includes means separate from the translation of the target instruction.

Claim 10. Computer implemented software means as claimed in Claim 8 in which the means for testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made is a part of the translation of the target instruction.

Claim 11. Computer implemented software means as claimed in Claim 8 which includes means for copying a memory address of a target instruction when a translation of the target instruction is made and linked to an earlier translation.

Claim 12. Computer implemented software means as claimed in Claim 8 which includes

means for copying a memory address of a target instruction when a translation of the target instruction is made, and

means for storing the memory address of a target instruction for comparison with a memory address of a target instruction to be executed.

Claim 13. Computer implemented software means as claimed in Claim 8 which includes means for executing the translation without testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made if the test can be safely eliminated.

Claim 14. Computer implemented software means as claimed in Claim 8 which includes means for executing the translation without testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made if the memory addresses are on the same memory page.

Abstract of the Disclosure

In a computer which translates instructions from a target instruction set to a host instruction set, a method for determining validity of a translation of a target instruction linked to an earlier translation including the steps of testing a memory address of a target instruction to be executed against a copy of the memory address of the target instruction from which a translation of the target instruction was made, executing the translation if the addresses compare, and generating an exception if the addresses do not compare.

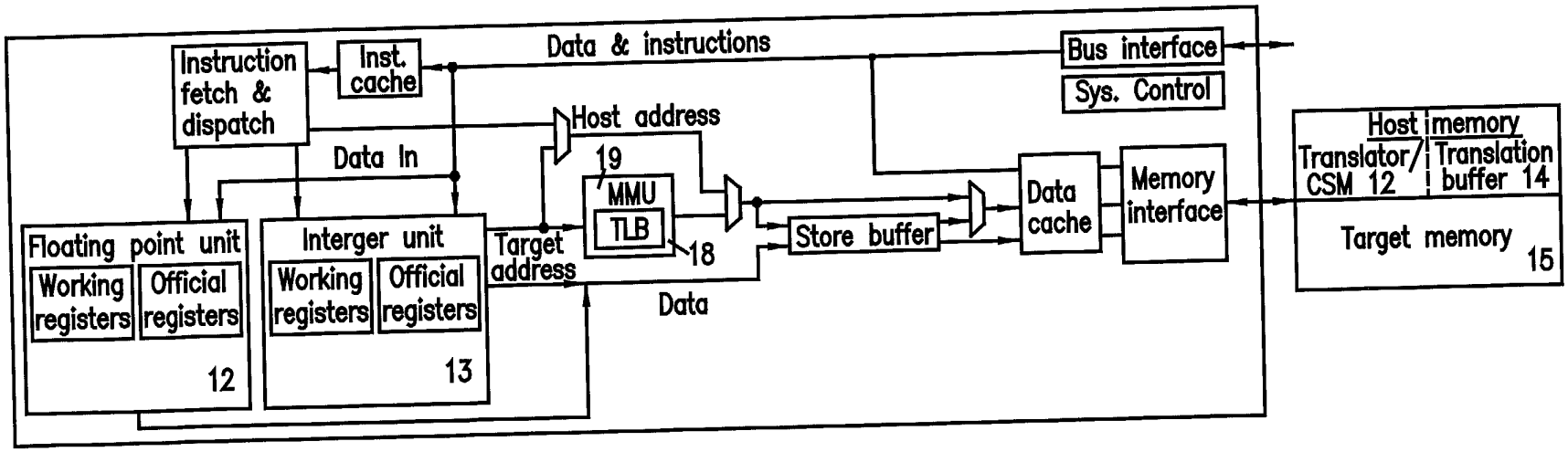
Instruction fetch & dispatch
Inst. cache
Data & instructions
Bus interface
Sys. Control
Host address
19
Data In
MMU
TLB
18
Floating point unit
Working registers
Official registers
12
Interger unit
Working registers
Official registers
13
Target address
Data
Store buffer
Data cache
Memory interface
Host memory
Translator/ CSM 12
Translation buffer 14
Target memory
15

Figure 1

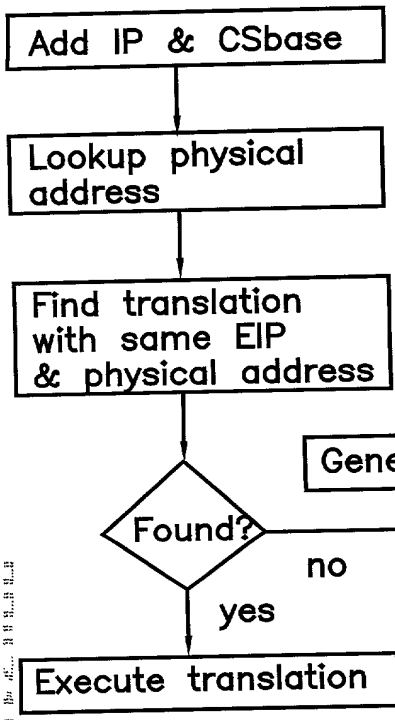

Figure 2

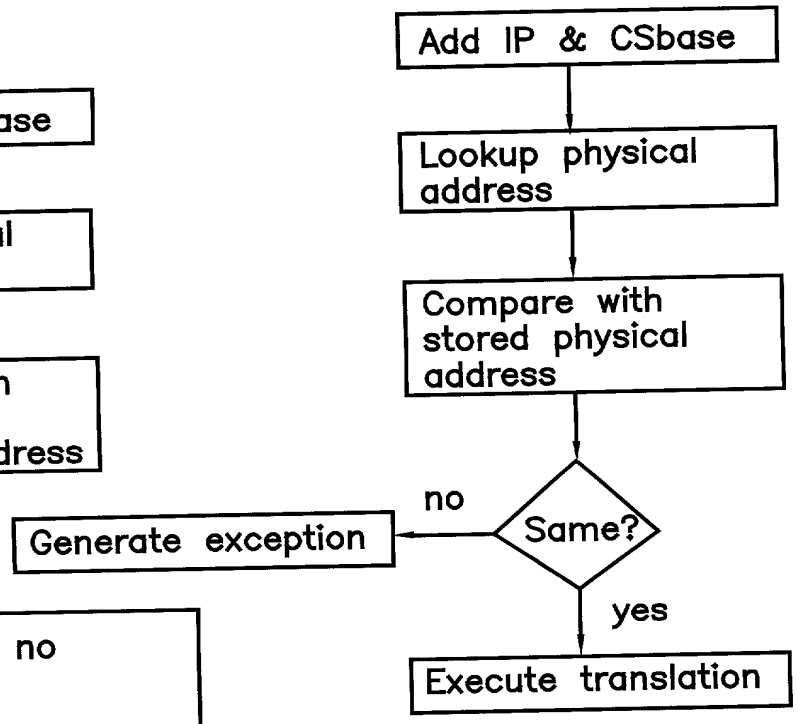

Figure 3

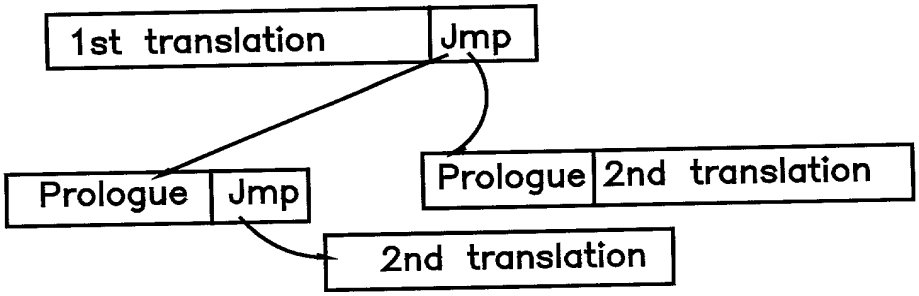

Figure 4

Attorney's Docket No.: Trans18 Patent

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

INTERPAGE PROLOGUE TO PROTECT VIRTUAL ADDRESS MAPPINGS

the specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby appoint Stephen L. King, Reg. No. 19,180; with offices located at 30 Sweetbay Road, Rancho Palos Verdes, California 90275, telephone (310) 377-5073, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of

Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor Robert Bedichek

Inventor's Signature _____ Date 17 Dec 1999

Residence Palo Alto, California (City, State) Citizenship U.S.A. (Country)

Post Office Address 2951 South Court
Palo Alto, California 94306

Full Name of Second/Joint Inventor David Keppel

Inventor's Signature _____ Date _____

Residence Seattle, Washington (City, State) Citizenship U.S.A. (Country)

Post Office Address 6852 19th Ave NE
Seattle, Washington 98115

Full Name of Third/Joint Inventor John Banning

Inventor's Signature _____ Date 16 DEC 99

Residence Sunnyvale, California (City, State) Citizenship U.S.A. (Country)

Post Office Address 808 Ticonderoga Drive
Sunnyvale, CA 94087